UDACITY

PROJECT SPECIFICATION

# Deploying a Sentiment Analysis Model

## Files Submitted

| CRITERIA | MEETS SPECIFICATIONS |
| --- | --- |
| Submission Files | The submission includes all required files, including notebook, python scripts, and html files.<br><br>Make sure your submission contains:<br><br>• The `SageMaker Project.ipynb` file with fully functional code, **all code cells executed and displaying output**, and **all questions answered**.<br>• An HTML or PDF export of the project notebook with the name `report.html` or `report.pdf`.<br>• The `train` folder with all provided files and the **completed** `train.py`.<br>• The `serve` folder with all provided files and the **completed** `predict.py`.<br>• The `website` folder with the **edited** `index.html` file. |

## Preparing and Processing Data

| CRITERIA | MEETS SPECIFICATIONS |
| --- | --- |
| **Question**: What does review_to_words do? | Answer describes what the pre-processing method does to a review. |

| CRITERIA | MEETS SPECIFICATIONS |
| --- | --- |
| | |
| Create a word dictionary | The `build_dict` method is implemented and constructs a valid word dictionary. |
| **Question:** What are the five most frequently appearing words? | Notebook displays the five most frequently appearing words. |
| **Question:** Understanding `preprocess_data` and `convert_and_pad_data` | Answer describes how the processing methods are applied to the training and test data sets and what, if any, issues there may be. |

### Build and Train the PyTorch Model

| CRITERIA | MEETS SPECIFICATIONS |
| --- | --- |
| Writing the training method | The train method is implemented and can be used to train the PyTorch model. |
| Training the model | The RNN is trained using SageMaker's supported PyTorch functionality. |

### Deploy the Model for Testing

## Deploy the Model for Testing

| CRITERIA | MEETS SPECIFICATIONS |
| --- | --- |
| | |
| Deploy the trained model | The trained PyTorch model is successfully deployed. |

## Use the Model for Testing

| CRITERIA | MEETS SPECIFICATIONS |
| --- | --- |
| **Question:** How does this model compare to the XGBoost model? | Answer describes the differences between the RNN model and the XGBoost model and how they perform on the IMDB data.<br><br>Make sure your answer includes:<br><br>• The comparison between the two models<br>• Which model is better for sentiment analysis |
| Process the test review | The test review has been processed correctly and stored in the `test_data` variable. The test_data should contain two variables: review_len and review[500]. |
| Writing inference code | The `predict_fn()` method in `serve/predict.py` has been implemented.<br><br>• The predict script should include both the data |

| CRITERIA | MEETS SPECIFICATIONS |
| --- | --- |
| | processing and the prediction.<br>The processing should produce two variables: data_X and data_len. |
| | |

### Deploying the Web App

| CRITERIA | MEETS SPECIFICATIONS |
| --- | --- |
| The web app is deployed | The model is deployed and the Lambda / API Gateway integration is complete so that the web app works (make sure to include your modified `index.html`). |
| **Question:** Give an example review and response | The answer includes a screenshot showing a sample review and the prediction. |

## Suggestions to Make Your Project Stand Out!

(1) MAKE A BETTER WEB APP

The web app that you make in this project simply reports to the user whether the predicted sentiment was positive or negative. Can you think of a better web app that uses the same model?

(2) IMPROVE THE WEB APP APPEARANCE

The provided web app is very simple and there is plenty of room for improvement if you wish to stretch your web developer skills.

(3) IMPROVE THE MODEL

The model chosen here is a straightforward RNN with a single hidden layer. There are many different model architectures that you could try to see if they improve the results.